Semi-supervised EM & Weak-Supervision in Anomaly Detection
CS229 Project Report

Minakshi Mukherjee, Suvasis Mukherjee: (adaboost, suvasism) @stanford.edu; adaboost.one@gmail.com

I. *Abstract*

Anomaly Detection from an unlabeled high dimensional dataset is a challenge in an unsupervised setup. Building models with a k-dimensional dataset(out of n-dimensional original feature set) under the assumption that the low k-dimensional subspace captures the variability in the data might affect the accuracy of model learning in the later phase. We analyzed NSL-KDD20% dataset[1], an improved version of UCI archive KDDCUP'99 network anomaly data using unsupervised and semi-supervised Expectation Maximization(EM) algorithms and observed how the choice of hyperparameters influences accuracy of the algorithm. In real world, intrusion detection and identification of security vulnerabilities within a network is an unsupervised problem. Since the ground truth(normal or attack) is available in this known dataset, the accuracy of different learning methods is compared against the ground truth. Apart from EM algorithm, we demonstrated weak supervision technique to label data with a few labelling functions by using snorkel[2]. Our results suggest that by using only a few labeled data Semi-Supervised EM(SSEM) uncovers distribution correctly, the log likelihood converges much faster and the mixture output maintains a stable data assignments. The results from Weak Supervision strongly suggest that a completely unlabeled dataset can be labeled with high accuracy using a sufficient number of labelling functions with accurate domain knowledge.

II. *Introduction*

Recent academic literature has demonstrated that current learning algorithms generally rely on labeled data and the algorithms cannot detect new or unknown intrusions in the network environment. Most of the intrusion detection algorithms include preset hard-coded rule finding functions provided by human experts. Also the number of features and the content of anomaly data depends heavily on the use case and the digital infrastructure of the organization. Most often, the large volume of network data is not labeled or labeled insufficiently and the accuracy of the labels can be questionable. We sought to study this problem in the context of Anomaly Detection.

Are semi-supervised algorithms for anomaly detection consistently converge under different circumstances? Do they help in detecting the outlier of a different attack type that the network dataset has not seen before?

We were particularly interested in this project due to the abundance of customer segmentation problems in all business domains where less than 1% customers are labeled for only one category, whereas inherent distribution of remaining 99% customers are unknown and there is a strong business demand to assign some labels or scores to the unlabeled data based on the behavior exhibited by the small subset of labeled data.

By completing this project, we hoped to make a novel and timely contribution to unsupervised or semi-supervised anomaly detection problem while learning the inherent nature of semi-supervised expectation maximization mixture model.

III. *Literature Review*

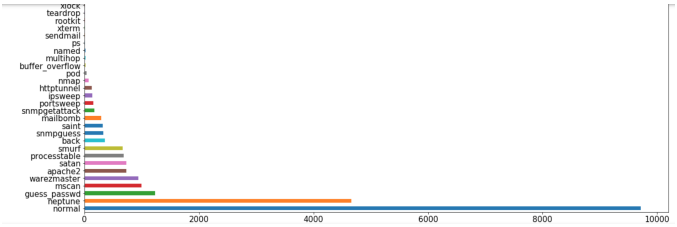
A network intrusion refers to any unauthorized activity on a digital network. Network intrusions often involve stealing valuable network resources and almost always jeopardize the security of networks and/or their data. There are a number of ways anomaly can be detected, we investigated unsupervised, semi-supervised and weak supervision learning methods. There are subjectivity inherent in unsupervised learning[3][8]. Hence, unsupervised learning is hard to evaluate. However, when there is no label information available, this is one of the ways to assign labels. Newer technologies, like LSTM Neural Networks, is useful in unsupervised and Semi-supervised Anomaly Detection. However, the performance of semi-supervised methods depends on sensitivity of the amount of labeled and unlabeled data, and performance can degrade substantially when the unlabeled dataset contains out-of-distribution examples. Unless the learner is absolutely certain there is some non-trivial relationship between labels and the unlabeled distribution, semi-supervised learning cannot provide significant advantages over unsupervised learning, SSL is no more than a constant factor better than SL for any unlabeled distribution[5]. Finally, use of Weak supervision [7] to learn the Structure of Generative Models without Labeled Data, requires domain knowledge to develop labeled functions and this may introduce bias. It is very interesting to see real world examples of mixture models in Natural language understanding, applications of drug discovery etc.[9][10]

IV. *Data Set and Features*

Our unsupervised and semi-supervised models are trained on NSL-KDD20 dataset[1], an improved version of UCI archive KDDCUP'99 network anomaly synthetic data that has a wide variety of intrusions simulated in a military network environment. There are four major categories: Denial of Service Attack (DoS), User to Root Attack (U2R), Remote

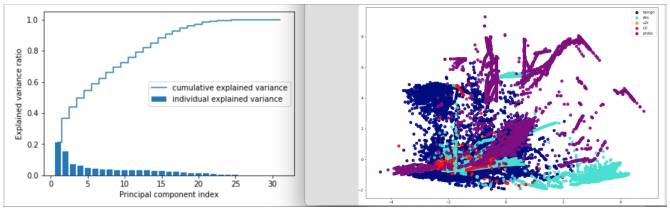
to Local Attack (R2L) and Probing Attack. The dataset comes with ground truth labels: (normal,attack).

Training data: 25,171 connection vectors with 41 features labeled as normal or attack with 4 attack categories. Training set has 24 attack types for those 4 attack categories.
Test data: 4,508 connection vectors with 41 features labeled as normal or attack with 4 attack categories. Test set has 38 attack types for those 4 attack categories. 41 features have 34 numerical variables and 7 categorical variables.
Training and test data came from different probability distributions. We noticed high class imbalance in different categories of attacks and the percentage of attack was close to 40% in both test and training data.



A. Feature Selection

We applied one hot encoding on 7 categorical variables and our total feature was 118. Since, scale of the distribution varied widely across the attributes, feature value standardization/normalization was applied across the variables. PCA of dimension 6 was used for feature selction as any dimension more than 6 resulted in singular matrix error during model building. PCA of dimension 6 accounted for 60% variability in the data.



In the real world anomaly detection problem, attack should not be more than 1-2%. Hence, in order to simulate the real world scenario, we used a specific data splitting strategy to run our algorithms.

V. *Models*

Models considered for the project

Unsupervised EM
Semi-Supervised EM
Weak Supervision

A. Data Selection for EM

Running Gaussian mixture model on multiple features can be computationally very intensive. We used hand written code to run the algorithms instead of commercial software. In order to run EM on our 16GB laptop, we split the training data per the data selection scheme described for each algorithm.

B. Unsupervised EM

The public UCI dataset is labelled. In order to run an unsupervised EM, we remove the label column and keep that aside for ground truth comparison. Hence, we have n training sets of network connection vectors $\{x^{(1)}, \ldots, x^{(n)}\}$ with 41 features without the label column. We learnt the model parameters μ, Σ, ϕ through the iterative E and M steps by maximizing a tractable lower bound of $p(x;\theta)$.

$$\ell_{\text{unsup}}(\theta) = \sum_{i=1}^{n} \log \sum_{z^{(i)}=1}^{k} p(x^{(i)}|z^{(i)};\mu,\Sigma)p(z^{(i)};\phi)$$

$z^{(i)}$ indicates which of the k Gaussians each $x^{(i)}$ had come from. As described earlier in the feature selection, instead of running the algorithm on all 41 features, we ran it on 6 attributes selected by PCA.

C. Unsupervised EM Data Selection

1. Randomly Split dataset X with (y label= "Normal") into k=20 disjoint subsets:$\{N_1, N_2, .., N_k\}$
2. Randomly Split dataset X with (y label = "Attack") into k=20 disjoint subsets which has representative proportion of 24 attack types in each of :$\{A_1, A_2, .., A_k\}$
3. Simulate real-world attack by taking 1% attack from each of $\{A_1, A_2, .., A_k\}$ and call it: $\{I_1, I_2, .., I_k\}$
4. Final data for Unsupervised Run:
S = $\{N_1 + I_1, N_2 + I_2, .., N_k + I_k\}$
5. For each j = 1..k
Run Unsupervised EM for different combinations of hyperparameters:
K= Number of Gaussians in the mixture model
I = Number of trials
ϵ = Convergence threshold
6. Final conclusion is based on the average results from all the hyperparameters

D. Semi-Supervised EM

In order to run Semi-supervised EM, we take n_1 training sets as labeled examples $\{(\tilde{x}^{(1)}, \tilde{z}^{(1)}), \ldots, (\tilde{x}^{(n2)}, \tilde{z}^{(n2)})\}$ where both x and z are observed, and we use n_2 training sets as unlabeled. We simultaneously maximized the marginal likelihood of the parameters using the unlabeled examples, and the full likelihood of the parameters using the labeled examples. The objective function to maximize is:

$$\ell_{\text{semi-sup}}(\theta) = \ell_{\text{unsup}}(\theta) + \alpha\ell_{\text{sup}}(\theta)$$

$$\mu^{(t+1)}, \Sigma^{(t+1)}, \phi^{(t+1)} := \arg\max_{\theta}$$

$$\left[\sum_{i=1}^{n_1}\left(\sum_{z_{(i)}=1}^{k} Q_i^{(t)}(z^{(i)}\log\frac{p(x^{(i)},z^{(i)};\theta)}{Q_i^{(t)}(z^{(i)})}\right)+\alpha\left(\sum_{i=1}^{n_2}(\tilde{x}^{(i)},\tilde{z}^{(i)};\theta)\right)\right]$$

E. Semi-Supervised EM Data Selection

1. Steps 1,2,3 are identical as of Unsupervised data selection.
2. Only additional step is to add 1% of labeled data in every final subset.
3. For each j = 1..k

Run Semi-Supervised EM for different combinations of hyperparameters:

K= Number of Gaussians in the mixture model
I = Number of trials
ϵ = Convergence threshold
α = Weight for the labeled examples

4. Final conclusion is based on the average results from all the hyperparameters.

F. Weak Supervision: Label training data using labeling api from snorkel[2]

In order to create noisy weak supervision sources to approximate the true labels, we removed the label column and kept that aside for ground truth comparison. Hence, we had n training sets of network connection vectors $\{x^{(1)},\ldots,x^{(n)}\}$ with 41 features without the label column. Instead of accessing labels, we have m weak supervision sources $\lambda_1,\lambda_2,...,\lambda_m$ that are functions from X to $Y\{1,0\}$ as noisy estimates of Y, these λ's are labelling functions(LF). In order to test weak-supervision, we took complete training data of 25,177 records without the y columns.Labeling functions (LFs) are used to encode domain knowledge and other supervision sources programmatically by using only one Rule Category –few Heuristic LFs were created by us:

There are 24 different attack types in the data and we handled only one Rule category based on the domain knowledge. By comparing the labels between λ's and the ground truth, we got **21%** the accuracy, this value is low because we did not have sufficient rules to apply.

However, this approach shows real promises, as when there are no labels available, we can do Probabilistic Labelling using Weak Supervision Sources.

rules: r2l_kdd_rules.pdf Formulation of heuristic rules for R2L attacks

1. If during an FTP session, large amount is data is sent from source as compared to destination then warezmaster attack can be concluded.

   (duration > 265) ∧ (protocol = tcp) ∧ (service = ftp ∨ ftp_data) ∧ (source_bytes > 265616) ∧ (destination_bytes = 0) Warezmaster Attack
2. If a guest has logged in through an FTP connection, and hidden directories are created then warezmaster attack can be concluded.

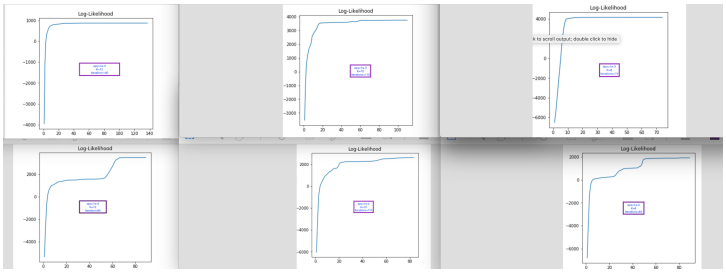
   (protocol = tcp) ∧ (service = ftp ∨ ftp_data) ∧ (hot > 0) ∧ (hot <= 2) ∧ (is_guest_login = 1) Warezmaster Attack
3. (duration > 265) ∧ (destination_bytes <= 688) ∧ (is_guest_login = 1) Warezmaster Attack
4. If a user, during an FTP session, triggers notably many hot indicators to be set in a small duration of time then the user maybe downloading software from the server

   (duration < 5) ∧ (protocol = tcp) ∧ (service = ftp ∨ ftp_data) ∧ (logged_in = 1 ∨ is_guest_login = 1) (hot > 25) Warezclient Attack
5. (duration <= 4685) ∧ (hot > 0) ∧ (hot <= 25) not Warezclient Attack
6. (source_bytes > 265616) ∧ (source_bytes <= 283618) Warezmaster Attack

VI. *Results from Unsupervised and SemiSupervised EM iterations*

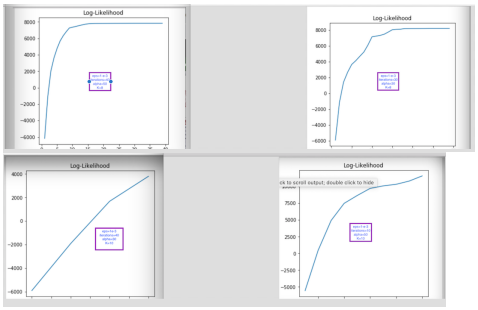
Best hyperparameter for unsupervised

log likelihood converged in 74 iterations for K=8,eps=1e-3



Best hyperparameter for SemiSupervised

log likelihood converged in 30 iterations for K=8,eps=1e-3 and alpha= 30



VII. *Model Accuracy and Discussion*

1.Did the EM algorithms discover the known structure of the data?

For a real-world unsupervised learning problem, this question is hard to answer, but here we compared our results against the ground truth labels.

We calculated the Detection rate and the False positive rate. The Detection rate is the number of intrusions detected by the system divided by the total number of intrusions presents in that test subset. The False positive rate is the total number of normal instances incorrectly classified as intrusions divided by the total number of normal instances in that test data.

Here is the summary of the model accuracy for unsupervised and semi-supervised:

	Accuracy	Detection_Rate (label_1)	Detection_Rate (label_2)	Detection_Rate (label_3)	Detection_Rate (label_4)
Unsupervised					
K=6, eps=1e-3	0.845	0.79	1	0.614	0
K=7, eps=1e-3	0.833	0.809	0.86	0	0
K=8, eps=1e-3	0.87	0.85	0.93	0.56	0
K=9, eps=1e-3	0.865	0.832	0.915	0	0
K=10, eps=1e-3	0.849	0.866	0.83	0.83	0
K=12,eps=1e-3	0.862	0.829	0.95	0.54	0
Semi-supervised					
K=8, eps=1e-3,alpha=30	0.912	0.882	0.91	0.59	0.33
K=10, eps=1e-3,alpha=30	0.89	0.91	0	0	0
K=12, eps=1e-3,alpha=30	0.898	0.892	0.83	0	0

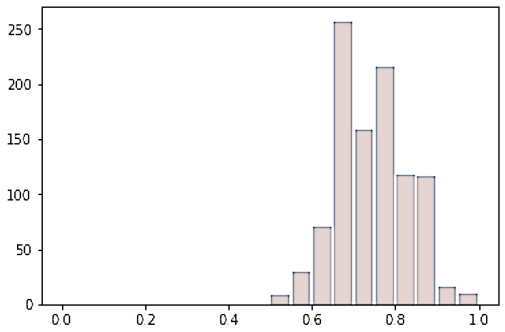
Unsupervised EM Accuracy: 85.4%
Semi-Supervised EM Accuracy: 90.0%

Also, our approach proved the following facts consistently for unsupervised EM and semi-supervised EM:

Log likelihood of Semi-supervised EM(SSEM) converges much faster
Additional labels helps SSEM to uncover distribution correctly

2.Did the EM algorithms provide stable cluster-to-labels matching?

We did **250** trials of our algorithm. Accuracy of unsupervised EM varied significantly as shown below:



However,accuracy of Semi-Supervised EM stayed in the range 86%-91% with the proper choice of the regularization parameter α.

To summarize, semi-supervised EM with just a few labeled examples(2%) provided a tremendous boost to anomaly detection with a proper selection of regularization parameter α.

VIII. $Limitations of EM$

Semi-supervised EM provides some stability due to the inclusion of a number of labeled examples, however unsupervised EM suffers from instability due to the heuristics of choosing number of Gaussian mixture components. Also, there is no precise definition of ”overfitting” or ”underfitting” for unsupervised learning problems. We cannot provide hard metrics like AUC to compare models across different families.

α too large: Log Likelihood of Semi-Supervised EM converges
too fast providing inaccurate cluster-to-label agreement
α too small: Log Likelihood of Semi-Supervised EM converges
after many iterations, however cluster-to-label agreement remains stable

Hence, model selection is still a subjective task and depends primarily on the skill of a machine learning engineer or researcher. The mixtures discovered by EM learning algorithms might be counter-intuitive and might not line up with human intuition, however, it’s proven mathematically that unsupervised and semi-supervised EM algorithms find “deep” features that explains a lot of variance. Hence these are great tools in the absence of any labels or in the presence of only a handful of labels.

IX. $Future\ Enhancements$

We explored two major approaches of anomaly detection above: dimensionality reduction using PCA and cluster analysis using GMM. A Gaussian Mixture Model of K components is a multimodal distribution with K or fewer bumps. The number of bumps depend on the number of distinct local maxima in the p.d.f.s. If the bumps are sufficiently close together or if the weight of one class is zero or nearly so, then a K component mixture models might have fewer than K bumps. This flexible structure of the mixture model makes it very suitable for modeling data, where there might be more than one region on high density. In spite of all these advantages, it is extremely computationally intensive to run GMM on a large dataset with many features. It also poses a difficult issue when there are a lot of categorical variables in the data as EM model requires all numeric inputs. We like to develop a robust data stratification algorithm that can be applied to a large dataset to take a representative sample so that EM algorithm can be run on those smaller subsets, instead of running semi-supervised EM on the entire dataset. While researching on NSL-KDD data(1], we did not find any paper that explored the anomaly detection problem on that data using unsupervised or semi-supervised algorithm. We provided a number of good findings and we like to take it to the next level, so that it can be executed at scale. . Also, Deep Semi-Supervised Anomaly Detection[6] is an interesting approach to consider for future work.

X. $Team\ Member\ Contribution$

Both of us contributed equally in researching and deciding on project topic, writing project report, developing small programs for milestone, executing and testing algorithms, writing posters, giving presentations and finally finishing up the project report.

We are very excited in successful completion of our project that demonstrated the power of semi-supervised EM and showed future promises for weak supervision labeling functions.

XI. *Github link*

https://github.com/suvasis/cs229_2019

REFERENCES

[1] https://www.unb.ca/cic/datasets/nsl.html
nsl-kdd

[2] https://github.com/snorkel-team/
snorkel

[3] https://arxiv.org/pdf/1710.09207.pdf
Unsupervised and Semi-supervised AnomalyDetection with LSTM Neural Networks

[4] https://arxiv.org/pdf/1804.09170.pdf
Realistic Evaluation of Deep Semi-SupervisedLearning Algorithms

[5] http://www.cs.toronto.edu/~tl/papers/lumastersthesis.pdf
Fundamental Limitations of Semi-Supervised Learning

[6] https://arxiv.org/pdf/1906.02694.pdf
Deep Semi-Supervised Anomaly Detection

[7] https://arxiv.org/pdf/1703.00854.pdf Learning the Structure of Generative Models without Labeled Data

[8] https://openreview.net/pdf?id=BJJLHbb0-
deep auto encoding with gmm

REAL WORLD EXAMPLES

[9] https://openreview.net/pdf?id=rygDeZqap7
Semi-supervised Ensemble Learning with Weak Supervisionfor Biomedical Relation Extraction

[10] https://arxiv.org/pdf/1906.10343.pdf
Exploring Self-Supervised Regularization for Supervised and Semi-Supervised Learning